

# Lesson 04

# 客户端 UI 编程

## 事件机制

当界面一件事情发生时，会执行一段代码

### 案例

鼠标划入

文字更改

屏幕尺寸更改

---

## 坐标

### 使用相对坐标计算

需求：

一个组件跟屏幕一样宽，左右两边各留 20 px

错误的做法：

```
1  label.setLayoutX(20);
2  label.setLayoutY(20);
3  label.setPrefWidth(760);
4  label.setPrefHeight(40);
```

正确的做法：

```
1  int margin = 20;
2  label.setLayoutX(margin);
3  label.setLayoutY(margin);
4  label.setPrefWidth(pane.getPrefWidth() - margin * 2);
5  label.setPrefHeight(40);
```

### 托盘效应

可以将组件放到一个容器组件中，组件此时的坐标会相对于容器

图示



Window



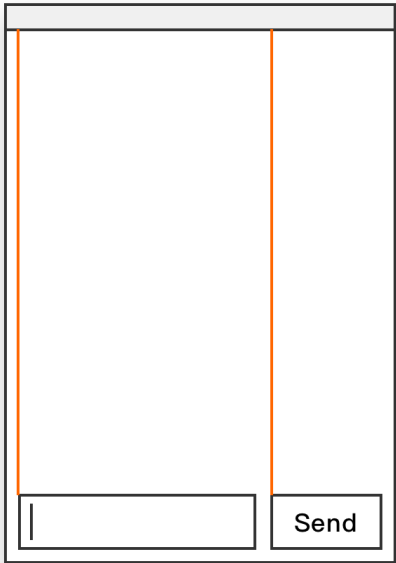
Label



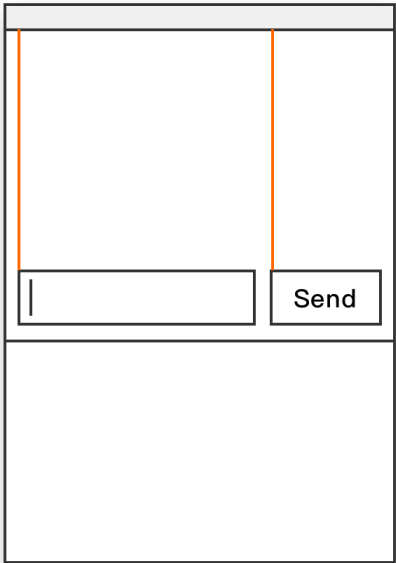
TextField



Button



键盘弹出



图示



Window



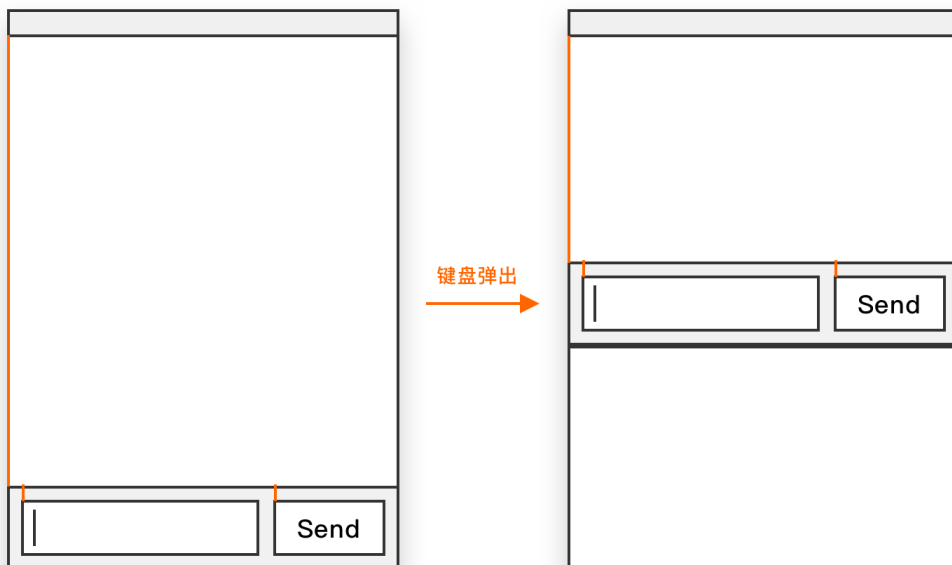
Label



TextField



Button



## 颜色

### 颜色显示机理

RGB

红绿蓝 三个光组成一个像素

### 颜色值

一般用 6 位 16 进制表达，比如

#FF0000 代表大红

#000000 代表黑

### API 拼凑

尝试寻找 API 设置 pane 的背景颜色

---

## 其它高级技术讨论

### 美化

贴图：

```
button.setBackground(new Image("res/images/btn.png"));
```

绘图：

```
canvas.drawCircle(centerPoint, radius, startAngle, endAngle)
```

### 搭建界面

XML：

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<Pane prefHeight="200" prefWidth="100" >
    <children>
        <Label prefHeight="100" prefWidth="100" layoutX="10" layoutY="10">label title</Label>
    </children>
</Pane>
```

拖拽：

拖拽程序，保存下来为 XML

### 布局

尺寸变了怎么办？

手工布局：

监听尺寸改变事件，  
重新设置组件的坐标数据

自动布局技术：

描述宽度是怎么算出来的

## 动画

### 手工更改：

设置一个计时器，  
监听计时器的事件  
重新设置组件的坐标数据

### 高级 API：

```
View.animate(1.0, {  
    button.x = 100;  
})
```

## 状态机

下载？  
尚未下载  
正在下载  
下载完成  
基于不同的状态，点击按钮会有不同的反应

---

# 持久化

## Persistence

## 计算机结构

CPU - Memory - Disk

### Disk：

持久化存储数据  
关机之后 依然在

### Memory：

存储数据  
程序运行时，任何一个变量都会开在内存里  
关机之后，就没了

### CPU：

用来计算

1 + 2 -> 3

## 持久化

关了还能用

文件

网络 发到服务器 服务器存储到文件里

---

## 文件操作

### 文件基本操作

File

### 文件读写

Apache Common IO

---

## 序列化

Serialization

### 问题

对象在内存里 是网状结构

文件 是1维的二进制

网络 是1维的二进制流

### 序列化 / 反序列化

Serialization / Deserialization

对象 -> 二进制

二进制 -> 对象

拆毛衣 / 织毛衣

分析：

优点：序列化之后的文件大小比较小

缺点：可读性非常低

## 字符串作为中介的序列化

想法：

对象 -> 字符串 -> 二进制

分析：

字符串 -> 二进制：

有现成的解决方案

会有字符集的问题

对象 -> 字符串：

可以写函数去拼接字符串